

**CIRCUIT COMPONENT FOR SYSTEM LSI WRITTEN IN HARDWARE
DESCRIPTION LANGUAGE, VERIFICATION METHOD THEREFOR,
VERIFICATION SUPPORT CIRCUIT AND MANUFACTURING METHOD FOR
SYSTEM LSI**

Publication number: JP2002288255

Publication date: 2002-10-04

Inventor: KATAYAMA ISAO

Applicant: TOKYO SHIBAURA ELECTRIC CO

Classification:

- international: G01R31/28; G06F17/50; H01L21/82; G01R31/28;
G06F17/50; H01L21/70; (IPC1-7): G06F17/50;
G01R31/28; H01L21/82

- European: G06F17/50C3; G06F17/50D

Application number: JP20010087045 20010326

Priority number(s): JP20010087045 20010326

Also published as:

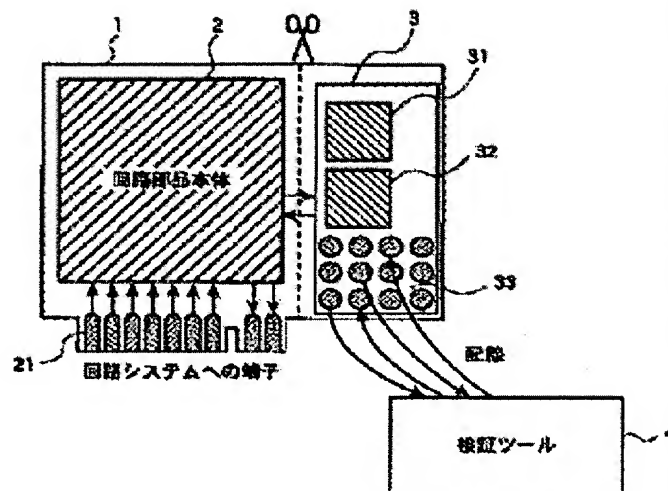
US7197731 (B2)
US2002138814 (A1)
KR20020076134 (A)
CN1378286 (A)
TW221310B (B)

more >>

Report a data error here

Abstract of JP2002288255

PROBLEM TO BE SOLVED: To provide circuit components for system LSI where the provided circuit components can be easily verified on the user side. **SOLUTION:** In the device, a verification support circuit, written in the hardware description language, is provided for a circuit component body which is written in hardware description language provided with a particular circuit function connected, so that it can be separated from the circuit component body by the provider of the circuit component and also connected to an interior of the circuit of the circuit component body, so that no adverse effects are imparted to the operation of the circuit component body itself, even when the connection with the circuit component body is cut off.



Data supplied from the esp@cenet database - Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-288255

(P2002-288255A)

(43) 公開日 平成14年10月4日 (2002. 10. 4)

(51) Int.Cl. ⁷	識別記号	F I	テマコード* (参考)
G 0 6 F 17/50	6 6 4	G 0 6 F 17/50	6 6 4 K 2 G 1 3 2
	6 5 4		6 5 4 K 5 B 0 4 6
	6 5 6		6 5 6 B 5 F 0 6 4
G 0 1 R 31/28		G 0 1 R 31/28	F
H 0 1 L 21/82		H 0 1 L 21/82	C
審査請求 未請求 請求項の数 9 O L (全 14 頁)			

(21) 出願番号 特願2001-87045(P2001-87045)

(22) 出願日 平成13年3月26日 (2001. 3. 26)

(71) 出願人 000003078

株式会社東芝

東京都港区芝浦一丁目1番1号

(72) 発明者 片山 功

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝マイクロエレクトロニクスセンター内

(74) 代理人 100083161

弁理士 外川 英明

Fターム(参考) 2G132 AA14 AB02 AC11 AK11 AL09
AL12

5B046 AA08 BA03 CA06 JA05

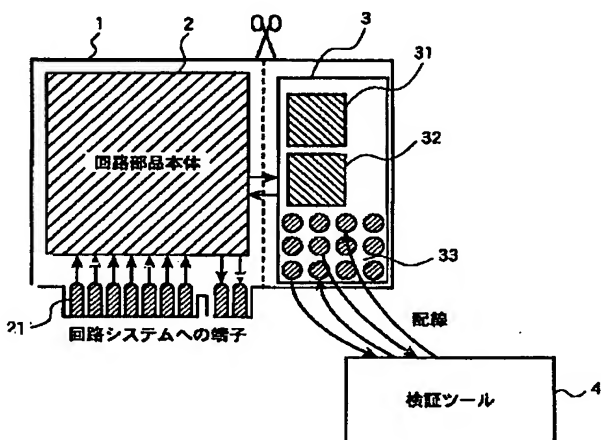
5F064 DD04 HH06 HH08 HH09 HH10
HH12

(54) 【発明の名称】 ハードウェア記述言語で記述されたシステムLSIの回路部品、及びその検証方法、検証支援回路、システムLSIの製造方法

(57) 【要約】

【課題】 提供された回路部品を使用者側にて容易に検証できるようなシステムLSIの回路部品を提供する。

【解決手段】 特定の回路機能を有する、ハードウェア記述言語で記述された回路部品本体に対し、回路部品の提供者によって回路部品本体と切り離し可能に接続されていると共に、回路部品本体との接続が切断された場合でも回路部品本体単体での動作に影響を与えないように回路部品本体の回路内部と接続されているハードウェア記述言語で記述された検証支援回路を設けるようにした。



【特許請求の範囲】

【請求項1】 特定の回路機能を有する回路であり、当該回路への入出力端子を備えたハードウェア記述言語で記述された回路部品本体と、

前記回路部品本体の回路内部に接続され、前記回路部品本体の回路内部の動作又は状態を表す信号を出力するためのハードウェア記述言語で記述された検証用出力端子とを備え、

前記検証用出力端子は、前記回路部品本体と切り離し可能に接続されていると共に、前記回路部品本体との接続が切り離された場合でも前記回路部品本体単体での動作に影響を与えないように前記回路部品本体と接続されていることを特徴とするハードウェア記述言語で記述されたシステムLSIの回路部品。

【請求項2】 前記回路部品本体に接続され、前記回路部品本体単体での動作を可能にするよう前記回路部品本体へ供給する動作クロック及びエミュレーション用信号を生成するためのハードウェア記述言語で記述された駆動回路を備えたことを特徴とする請求項1記載のハードウェア記述言語で記述されたシステムLSIの回路部品。

【請求項3】 前記検証用出力端子から前記回路部品本体の動作又は状態を表す信号を外部へ引き出すためのハードウェア記述言語で記述されたモニタ接続回路を備えたことを特徴とする請求項1または請求項2記載のハードウェア記述言語で記述されたシステムLSIの回路部品。

【請求項4】 特定の回路機能を有する回路であり当該回路への入出力端子を備えた回路部品本体と、前記回路部品本体と切り離し可能に接続されると共に前記回路部品本体との接続が切り離された場合でも前記回路部品本体単体での動作に影響を与えないよう前記回路部品本体の回路内部と接続される検証用出力端子とが設けられたハードウェア記述言語で記述された回路部品の単体、又は少なくとも1つの前記回路部品の単体を有するハードウェア記述言語で記述された回路システムをシミュレーション動作させるステップと、
前記回路部品本体の動作又は状態を表す信号を前記検証用出力端子を介してシミュレータの表示画面に表示出力させるステップとを備えたことを特徴とするハードウェア記述言語で記述されたシステムLSIの回路部品の検証方法。

【請求項5】 前記回路部品は前記回路部品本体を単体動作させるための動作クロック及びエミュレーション用信号を生成する駆動回路を有しており、前記回路部品単体をシミュレーション動作させるステップは、前記駆動回路にて生成された動作クロック及びエミュレーション用信号を前記回路部品本体に供給するステップを更に有することを特徴とする請求項4記載のハードウェア記述言語で記述されたシステムLSIの回路部品の検証方

法。

【請求項6】 入出力端子を備え特定の回路機能を有するハードウェア記述言語で記述された回路部品に対して接続／切り離しが可能な回路であり、前記回路部品内部の動作又は状態を表す信号を外部出力するための検証用出力端子を備えたことを特徴とするハードウェア記述言語で記述された検証支援回路。

【請求項7】 入出力端子を備え特定の回路機能を有するハードウェア記述言語で記述された回路部品に対して接続／切り離しが可能な回路であり、前記回路部品単体での動作を可能にするよう前記回路部品へ供給する動作クロック及びエミュレーション用信号を生成するための駆動回路を備えたことを特徴とするハードウェア記述言語で記述された検証支援回路。

【請求項8】 入出力端子を備え特定の回路機能を有するハードウェア記述言語で記述された回路部品に対して接続／切り離しが可能な回路であり、前記回路部品に接続された前記回路部品内部の動作又は状態を表す信号を外部出力するためのハードウェア記述言語で記述された検証用出力端子からの信号を前記回路部品外部へ引き出すためのモニタ接続回路を備えたことを特徴とするハードウェア記述言語で記述された検証支援回路。

【請求項9】 特定の回路機能を有する回路であり当該回路への入出力端子を備えた回路部品本体と、前記回路部品本体と切り離し可能に接続されると共に前記回路部品本体との接続が切断された場合でも前記回路部品本体単体での動作に影響を与えないよう前記回路部品本体の回路内部と接続される検証用出力端子を少なくとも有する検証支援回路とが設けられた、ハードウェア記述言語で記述された回路部品の単体、又は少なくとも1つの前記回路部品の単体を有するハードウェア記述言語で記述された回路システムをシミュレーション動作させると共に、前記回路部品本体の動作又は状態を表す信号を前記検証用出力端子を介してシミュレータの表示画面に表示出力させてシミュレーション動作検証を行うステップと、

前記シミュレーション動作検証にて正常動作した前記回路部品の単体又は回路システムから前記検証支援回路を切り離した回路記述に基づいて論理合成を行うステップと、

前記論理合成により得られた論理回路を基に回路の配置配線を行うステップと、

前記配置配線された回路の配置配線データに基づいて基板上に実回路を構築するステップとを有することを特徴とするシステムLSIの製造方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、システムLSIの回路部品に関するものであり、知的労働の産物である回路設計資産（IP: Intellectual Property）に関す

る。具体的には、自己検証機能を備えたシステムLSIの回路部品の構造、検証方法、検証支援回路、及びシステムLSIの製造方法に関する。

【0002】

【従来の技術】VHDLやVerilog（登録商標）—HDLに代表される近年のハードウェア記述言語やシミュレーション技術、回路合成技術の向上により、1つのLSI（大規模集積回路：large scale integrated circuit）のチップ上に電子回路システムを実現するシステムLSI（System on Chip）の設計者の多くは、回路システムの動作仕様をハードウェア記述言語で記述し、この記述がシステム要求機能を満たしているかどうかをシミュレーションCADを用いて検証した後、回路合成ツール（論理合成CAD）を用いてこの記述に対応する実際のゲートレベルの論理回路の合成を行っている。また、最近では回路設計期間短縮の要求が強いことから、設計者が回路システムを一から設計することが少なくなり、設計する回路システムを機能毎に分割して、必要に応じて市場で流通しているそれぞれの専門分野のベンダ（販売業者／プロバイダ）から提供されるハードウェア記述言語レベルでの回路設計データを回路部品として調達し、それらを組み合わせて回路システムを構築することが多くなってきている。

【0003】その一方で、システム規模の拡大や機能の高度化に伴い、組み合わせる回路部品の種類は増える傾向にあり、また、回路部品を提供するベンダ数も増加してきていることから、システム設計者側としては、調達した回路部品ごとの動作検証基準や回路の品質などのバラツキを無視することができなくなっている。また、回路部品の高度化とベンダ間における競争とから、回路部品の内部情報の隠蔽化／ブラックボックス化が進み、システム設計者側では調達した回路部品の品質検証が非常に困難な状況になってきている。

【0004】LSIの設計データは製品化後の修正が極めて困難であることから、製品化前の動作検証が非常に重要となる。調達した回路部品単体、及び複数の回路部品の組み合わせからなる回路システム全体が正常に動作するかどうかは、製品化前の回路設計段階においてシミュレーション等で検証しておく必要がある。シミュレーションによって正しい動作が得られなかった場合は、この段階で異常の原因となった回路を特定して修正を施す必要がある。

【0005】通常、回路部品は提供者側（ベンダ；vendor）によってその動作が保証されているが、上述した理由から使用者側（システム設計者側）でも独自に検証を行いたいという要求が強い。従来、調達した回路部品単体の検証を使用者側で行う場合には、回路部品提供者が公開する限られた範囲の仕様（通常は入出力端子の仕様のみ）と、その回路部品が動作する外部条件とから使用者側にて独自に検証用駆動回路を作成し、シミュレータ

ーにて回路部品を単体で動作させてその動作を検証していた。この時、回路部品の動作に異常が認められると問題の発生源を特定して原因究明を行うことになるが、そのためには回路部品を一から内部解析する必要がある。しかしながら、回路部品の高度化と内部情報の隠蔽化により、回路部品の内部解析、並びに回路部品内部での異常の原因箇所の特定は非常に困難な状況にある。また、実際には異常の原因が独自作成した検証用駆動回路側にあり、回路部品そのものには異常がなかったにも関わらず、結果的にその特定のために回路部品の内部解析を行ってしまうということも多い。また、システムとして組み上がった段階で動作異常が確認された場合には、複数の回路部品の中から異常を起こした回路部品の特定、及びその原因究明を行うために回路部品個々の内部解析を行わなければならないが、前述と同様の理由から回路部品の内部解析を行うことは困難であり、回路部品間の異常の特定、並びにその対策にかなりの時間を割かなければならなかった。

【0006】

【発明が解決しようとする課題】上述したように、回路部品単体や回路システム全体の検証を行う場合、その動作に異常が発生した場合には回路部品個々の内部解析を行うことが必要不可欠である。しかしながら、最近の回路部品の高度化と内部情報の隠蔽化により、回路部品の内部解析、並びに回路部品内部での異常の原因箇所の特定は非常に困難な状況となってきている。また、回路部品の内部解析を行うにしても、その複雑さから膨大な時間と労力がかかり、回路部品調達の当初の目的であった設計期間の短縮化の効果が十分に得られないと言う問題も生じてきている。

【0007】本発明はこのような問題点に鑑みてなされたものであり、その目的は、提供された回路部品を使用者側にて容易に検証できるようなシステムLSIの回路部品の構造、検証方法、検証支援回路、及びシステムLSIの製造方法を提供することにある。

【0008】

【課題を解決するための手段】前記課題を解決するために、本発明のハードウェア記述言語で記述されたシステムLSIの回路部品は、特定の回路機能を有する回路であり、当該回路への入出力端子を備えたハードウェア記述言語で記述された回路部品本体と、前記回路部品本体の回路内部に接続され、前記回路部品本体の回路内部の動作又は状態を表す信号を出力するためのハードウェア記述言語で記述された検証用出力端子とを備え、前記検証用出力端子は、前記回路部品本体と切り離し可能に接続されていると共に、前記回路部品本体との接続が切り離された場合でも前記回路部品本体単体での動作に影響を与えないように前記回路部品本体と接続されていることを特徴とする。

【0009】また、本発明のハードウェア記述言語で記

述されたシステム L S I の回路部品の検証方法は、特定の回路機能を有する回路であり当該回路への入出力端子を備えた回路部品本体と、前記回路部品本体と切り離し可能に接続されると共に前記回路部品本体との接続が切り離された場合でも前記回路部品本体単体での動作に影響を与えないよう前記回路部品本体の回路内部と接続される検証用出力端子とが設けられたハードウェア記述言語で記述された回路部品の単体、又は少なくとも 1 つの前記回路部品の単体を有するハードウェア記述言語で記述された回路システムをシミュレーション動作させるステップと、前記回路部品本体の動作又は状態を表す信号を前記検証用出力端子を介してシミュレータの表示画面に表示出力させるステップとを備えたことを特徴とする。

【 0 0 1 0 】また、本発明のハードウェア記述言語で記述された検証支援回路は、入出力端子を備え特定の回路機能を有するハードウェア記述言語で記述された回路部品に対し接続／切り離しが可能であり、前記回路部品内部の動作又は状態を表す信号を外部出力するための検証用出力端子、又は前記回路部品単体での動作を可能にするよう前記回路部品へ供給する動作クロック及びエミュレーション用信号を生成するための駆動回路、又は前記回路部品に接続された前記回路部品内部の動作又は状態を表す信号を外部出力するためのハードウェア記述言語で記述された検証用出力端子からの信号を前記回路部品外部へ引き出すためのモニタ接続回路を備えたことを特徴とする。

【 0 0 1 1 】また、本発明のシステム L S I の製造方法は、特定の回路機能を有する回路であり当該回路への入出力端子を備えた回路部品本体と、前記回路部品本体と切り離し可能に接続されると共に前記回路部品本体との接続が切断された場合でも前記回路部品本体単体での動作に影響を与えないよう前記回路部品本体の回路内部と接続される検証用出力端子を少なくとも有する検証支援回路とが設けられた、ハードウェア記述言語で記述された回路部品の単体、又は少なくとも 1 つの前記回路部品の単体を有するハードウェア記述言語で記述された回路システムをシミュレーション動作させると共に、前記回路部品本体の動作又は状態を表す信号を前記検証用出力端子を介してシミュレータの表示画面に表示出力させてシミュレーション動作検証を行うステップと、前記シミュレーション動作検証にて正常動作した前記回路部品の単体又は回路システムから前記検証支援回路を切り離した回路記述に基づいて論理合成を行うステップと、前記論理合成により得られた論理回路を基に回路の配置配線を行うステップと、前記配置配線された回路の配置配線データに基づいて基板上に実回路を構築するステップとを有することを特徴とする。

【 0 0 1 2 】

【発明の実施の形態】以下、本発明の一実施形態につい

て図面を参照して説明する。

【 0 0 1 3 】図 1 は、本発明の一実施形態であるハードウェア記述言語からなる回路部品の模式図である。本実施の形態における回路部品 1 は、ハードウェア記述言語で記述される特定の回路機能（MPEG2 デコード、DSP、等）を有する回路部品本体 2 と検証支援回路 3 とから構成される。回路部品本体 2 は、他の回路システムと接続するための仕様が公開された入出力端子 2 1 を有する。回路部品本体 2 は、提供者により通常は特別のアルゴリズムや情報の隠蔽化処理が施されており、回路部品の利用者が得られる内部情報は前記入出力端子 2 1 程度であることが多い。

【 0 0 1 4 】検証支援回路 3 は、本発明の回路部品特有の構成であり、回路部品 1 として回路部品本体 2 と一緒に付加されて提供されるものである。検証支援回路 3 は、回路部品本体 2 単体での駆動を可能にするための駆動回路 3 1 と、回路部品本体 2 の動作状態を表す信号を出力するための検証用出力端子 3 3 と、検証用出力端子 3 3 から回路部品本体 2 の動作状態を表す信号を外部へ引き出すためのモニタ回路 3 2 とを有する。

【 0 0 1 5 】駆動回路 3 1 は、回路部品本体 2 に供給するためのクロックの生成や初期化信号、接続する回路システムとのインターフェースのエミュレーション（回路部品本体 2 への信号の擬似入出力）を行うものであり、回路部品本体 2 を単体で動作させられるよう回路部品 1 の提供者によって予め回路部品本体 2 と接続されている。また、モニタ回路 3 2 は、駆動回路 3 1 による回路部品本体 2 の単体動作時、または設計しようとする回路システムに組み込んだ際の動作時に、回路部品本体 2 の動作状況が検証用出力端子 3 3 を介して外部検証環境へ出力できるよう回路部品 1 の提供者によって提供される。検証用出力端子 3 3 は、回路部品本体 2 の内部に接続されており、検証時に外部検証ツール 4 などの検証環境に接続するための端子である。検証用出力端子 3 3 の仕様は、予め回路部品 1 の提供者によって公開されている。

【 0 0 1 6 】尚、検証支援回路 3 は、回路部品本体 2 と同様のハードウェア記述言語で記述されており、必要に応じて使用者により回路部品本体 2 との接続を切り離すことが可能である。回路部品本体 2 は、検証支援回路 3 との接続の切り離しによって回路部品本体 2 が異常動作することがないように回路記述されている。

【 0 0 1 7 】図 2 乃至図 4 は、図 1 に示した回路部品 1 のハードウェア記述言語による記述の一例である。図 5 及び図 6 は、検証用出力端子 3 3 の仕様に基づいて使用者によって作成されたハードウェア記述言語による検証ツール 4 の記述の一例である。

【 0 0 1 8 】以下、図 2 乃至図 4 を用いて回路部品 1 の一例について具体的な説明を行う。

《回路部品 1》<回路部品本体 2> [入出力端子 2 1]

図2乃至図4には、Compo23Aという名称のモジュールがハードウェア記述言語で記述されている。ここには、例えばMPEG2デコーダ等の特定の回路機能を有する回路部品本体2の回路動作記述（/* Circuit Discription */以下の記述部分；ただし回路部品本体の機能については本発明とは直接関係しないため、図面中では記述を省略している）の他、仕様の公開されている回路部品本体2の入出力端子21（in00sig、in01sig、.../out00sig、out01sig、...）、及び仕様の公開されていない回路部品本体2の内部信号線（Internal00Wire、Internal01Wire、...）が定義されている。

<検証用支援回路3> [検証用出力端子33]

このCompo23Aには、検証用出力端子33（oStatusCode、oConditionCode、...）が定義されると共に、検証用出力端子33と回路部品本体2との接続が記述されている。検証用出力端子33の仕様（例えば、端子oStatusCodeは内部バスのデータ入力可否の状態を表す／端子oActiveRatioはバスの稼働率の状態を表す／端子oChangeStateはoStatusCodeの変化の状態を表す／...等）については、提供者により使用者に公開されている。

【0019】以下に、検証用出力端子33と回路部品本体2との接続の具体例を示す。

（1）内部信号線の状態を直接出力するような接続の一例。

<記述例> “assign oStatusCode = Internal84Wire;”
（意味：端子oStatusCodeと内部信号線Internal84Wireとを接続する）

（2）複数の内部信号線間の状態に基づく演算結果を出力するような接続の一例。

<記述例> “assign oActiveRatio = Internal84Wire / Internal12Wire * 100;”
（意味：内部信号線Internal12Wireの状態に対する内部信号線Internal84Wireの状態の比率（%）を端子oActiveRatioに接続する）

（3）内部信号線間の状態比較結果に基づく接続の一例。

<記述例> “assign oChangeState = (Internal61Wire == Internal62Wire) ? 1'b0 : 1'b1;”
（意味：内部信号線Internal61Wireの状態と内部信号線Internal62Wireの状態とを比較し、一致した場合は端子oChangeStateに1 bit長binary表示の値0を接続し、不一致の場合は端子oChangeStateに1 bit長binary表示の値1を接続する）

（4）内部信号線の状態監視結果に基づく接続の一例。

<記述例> “assign oWaitCount = (Internal34Wire == 1'b0) ? Internal43Wire : Internal32Wire;”
（意味：内部信号線Internal34Wireの状態を監視し、1 bit長binary表示の値0と一致した場合は端子oWaitCountと内部信号線Internal43Wireとを接続し、不一致の場合は端子oWaitCountと内部信号線Internal32Wireとを接

続する）

（5）内部信号線の状態変動以前の状態を保持するレジスタ出力との接続の一例。

<記述例> “always @ (oChangeState) rPreviousData = Internal22Wire;” “assign oPreviousdata = rPreviousdata;”

（意味：端子oChangeStateの状態出力の変化毎に内部信号線Internal22Wireの状態が格納されるレジスタrPreviousDataの信号と端子oPreviousdataとを接続する）

このように、検証用出力端子33は、回路部品本体2の内部動作状態を表す信号と接続がされている。検証用出力端子33からは、検証時にオプション設定“Debug”が定義された場合に、上記のような接続に基づいて回路部品本体2内部からの信号が出力される。この接続については、回路部品1の提供者によって予め接続されており、また、この接続は接続が切断された場合であっても回路部品本体2単体の動作には影響を与えないよう考慮されている。

<検証用支援回路3> [駆動回路31]

図3及び図4には、Compo23Aという名称のモジュールに続いて、Driver23Aという名称のモジュールが記述されている。Driver23Aには、Compo23Aを単体で動作させるための駆動回路31が記述されており、動作クロックを生成し（/* Clock Driver */以下の記述部分）、回路部品外部のエミュレーション用の信号生成を行い（/* Reset & Singal Generator */以下の記述部分）、生成された信号を回路部品本体2の入力端子21へ入力する（/* Driving Terminals of Component */以下の記述部分）ための回路記述がされている。この駆動回路31は、検証時にオプション設定“StandAlone”が定義される場合（“`ifdef StandAlone”）にCompo23Aを駆動するような回路部品1の提供者によって予めCompo23Aと接続されている。

<検証用支援回路3> [モニタ接続回路32]

Driver23Aの中には、回路部品本体2の動作状況を検証用出力端子33を介して外部検証環境へ出力するためのモニタ回路32も記述されている。図4の記述例では、モニタ接続回路32は、検証時にオプション設定“ExternalMonitor”が定義された場合（“`ifdef ExternalMonitor”）に“DebugEnv”という名称の外部検証ツールを読み込み（“include “DebugEnv”;”）、外部検証ツールに作成されたモニタ回路を読み出す構成となっている。続いて、図5及び図6を用いて検証ツール4の一例について具体的に説明を行う。図5及び図6に示す検証ツール4（“DebugEnv”）では、検証時にCompo23Aに対して入力する信号データ名の記述（/* Input Singal Data */以下の記述部分）のあと、回路部品本体2の動作監視のためのモニタ回路（/* Signal Monitor */以下の記述部分）、回路部品1に与える検証条件の設定（/* Condition Settings for Debug */以下の記述部分）、回

回路部品1の状態監視のためのモニタ回路（/* Status Monitor */以下の記述部分）が記述されている。この検証ツール“DebugEnv”は、回路部品1の提供者により公開された検証用出力端子33の名称やその仕様を基に使用者によって作成（記述）されるものであり、回路部品1の検証時にオプション設定“ExternalMonitor”を定義することにより動作し、回路部品1の検証用出力端子33を介して得られる回路部品本体2内部の動作／状態データが、シミュレーション検証が行われるワークステーション等のコンピュータの表示画面に表示出力されるよう回路記述がなされている。

【0020】次に、本発明の一実施形態である回路部品1に対してシミュレーション検証を行う場合について説明する。

【0021】図7は、図1に示した回路部品1（1a）を使用して回路部品の導入から回路システムとして回路合成（論理合成）を行うまでの各段階の一例を示した模式図である。

【0022】図7aは、回路部品1a導入時に回路部品1aを単体駆動して回路部品本体2aの動作検証と評価を行う段階の一例を示した図であり、回路部品本体2aと検証支援回路3a、外部検証ツール4との関係を示している。ハードウェア記述言語で記述された回路部品1aの動作検証は、ハードウェア記述言語に対応するシミュレーションプログラムが動作する図示しないコンピュータ（ワークステーション等）にて行われる。

【0023】シミュレーション検証実行時に、オプション設定として“Debug”、“StandAlone”、“ExternalMonitor”が定義されると、回路部品1aに付加されている駆動回路31aからは回路部品本体2aに対して動作クロックや検証用エミュレーション信号が供給され、回路部品本体2aを単体動作させる。この時、回路部品本体2aの内部動作データ／状態データは、検証用出力端子33aを介してコンピュータの表示画面にモニタされ、正常動作の確認を行うことができる。必要があれば、この時得られたデータから回路部品本体2aの性能見積を行うことも可能である。この時点で正常動作確認が得られなかった場合は、回路部品本体2aが回路異常を有していると判断でき、回路部品本体2aの解析を行い、異常の原因を発見して異常の原因を取り除く。この異常の原因を発見するにあたっては、検証用出力端子33aから得られる回路部品2a内部の動作／状態データが役立つことになる。

【0024】図7bは、異なる機能を有する複数の回路部品1a、1b、1cを組み合わせるハードウェア記述言語で記述された回路システム5を構築した場合に、回路システム全体の検証と評価を行う段階の一例を示した模式図である。各回路部品1a、1b、1cは、回路システム5を形成するように各入出力端子21a、21b、21cを介してハードウェア記述言語で記述される回路

システム5と接続されている。また、必要に応じて回路システム5には検証ツール4との接続用の検証用出力端子53aを設けておいてもよい。各回路部品本体2a、2b、2cには検証支援回路3a、3b、3cが接続されており、内蔵のモニタ接続回路32a、32b、32cによって検証用出力端子33a、33b、33cを介して、各回路部品本体2a、2b、2cの内部動作／状態データが出力できるよう構成されている。

【0025】シミュレーション検証実行時に、オプション設定として“Debug”、“ExternalMonitor”が定義されると、回路システム5の動作における各回路部品本体2a、2b、2cの内部動作データ／状態データは、各検証用出力端子33a、33b、33cを介してコンピュータの表示画面にモニタされる。また、回路システム5に検証用出力端子53を設けている場合には、回路システム5全体としての動作状態をモニタすることも可能である。これらモニタされるデータは、システム全体の動作に異常が発生した場合には異常の原因となった回路部品の特定に利用される。各回路部品単体では動作に異常が無い場合でも、これら複数の回路部品を組み合わせた場合には特定の回路部品間で動作条件が衝突して正常動作しなくなることがあるが、各回路部品内部における動作／状態データを同時にモニタすることができるので、衝突の原因となった回路部品の特定が容易となり、その対策（回路修正）に掛かる時間を従来に比べて短縮することができる。

【0026】図7cは、回路システム5全体の検証終了後に、各回路部品1a、1b、1cから各検証支援回路3a、3b、3cを切り離した回路部品1'a、1'b、1'cにより構成された回路システム5'の一例を示す模式図である。論理合成を行う場合は、図7a、図7bの各段階にて検証済みの回路部品1'a、1'b、1'c（回路部品本体2a、2b、2c）が接続された回路システム5'を論理合成ツールへ入力することにより、検証支援回路3a、3b、3cが論理回路として反映されない回路システムを論理合成することができる。前述したように、本発明による回路部品では、回路部品本体と検証用支援回路との接続は接続を切断した場合でも回路部品本体の動作に影響を与えないように接続されている。したがって、動作検証済みの回路部品1から検証支援回路3を削除した場合でも、回路部品本体2の動作は保証される。

【0027】尚、回路部品1a、1b、1cから検証支援回路3a、3b、3cを削除するには、論理合成ツールへの入力において各回路部品に対するオプション設定“Debug”、“StandAlone”、“ExternalMonitor”を全て無効、あるいはオプション設定“Debug”、“StandAlone”、“ExternalMonitor”に関わる記述を全て無視または削除することにより実行でき、回路部品本体2a、2b、2cと検証支援回路3a、3b、3c（駆動回路

31a、31b、31c、モニタ接続回路32a、32b、32c、検証用出力端子33a、33b、33c)との接続は切断される。

【0028】このようにして論理合成された回路システムは、その後、配置配線処理を行った後、対応するマスクを生成し、生成したマスクに基づいてインプラ、エッチング等、所定の半導体製造工程を経てシリコン基板上にトランジスタや配線等の実論理回路を構築することによって、最終的にシステムLSIとして製造される。

【0029】図8は、本発明の回路部品に基づく回路部品の導入から検証、論理合成を経てシステムLSIに至るまでのシステムLSIの製造方法のフローチャートである。

【0030】ステップS1にて、設計者は市場に流通しているハードウェア記述言語によって記述された特定の機能を有する回路部品の購入／導入を行う。設計者は導入した回路部品に対して、予め回路部品本体に接続された検証支援回路である検証用出力端子からの出力をモニタしながらシミュレータにて回路部品単体の動作検証（シミュレーション）を行う（ステップS2）。動作検証の結果（ステップS3）、回路部品が正常動作しなかった場合には検証用出力端子からのモニタ出力を利用して異常の原因を発見すると共に異常原因の修正を行う（ステップS4）。

【0031】回路部品単体の動作に異常がなかった場合は、この回路部品を使用して回路システムの設計を行う（ステップS5）。回路システム設計後は、今度は回路システム全体の動作検証を行う（ステップS6）。この時、設計者は各回路部品に設けられている予め回路部品本体に接続された検証用出力端子からの出力をモニタする。動作検証の結果（ステップS7）、回路システムが正常動作しなかった場合は検証用出力端子からのモニタ出力を利用して異常の原因となった回路部品の特定を行うと共に、その対策（回路修正）を回路部品の提供元又は導入者側で行う（ステップS8）。

【0032】回路システムの動作に異常がなかった場合は、論理合成ツールに対して回路システムの記述を入力し、回路システムの記述の中から検証支援回路に関する記述を切り離した上で論理合成を行う（ステップS9）。本発明の回路部品の構造によれば、回路システムの記述のうち検証支援回路に関する記述を除いた回路のみを論理合成することが可能である。

【0033】論理合成された回路システムは、その後、配置配線処理（ステップS10）を行った後、対応するマスクを生成し（ステップS11）、生成したマスクに基づいてインプラ、エッチング等、所定の半導体製造工程（ステップS12）を経てシリコン基板上にトランジスタや配線等の実論理回路を構築することにより、最終的にシステムLSI（ステップS13）として製造される。

【0034】上述したシステムLSIの製造方法によれば、回路システムの動作検証にかかる時間が従来よりも短縮されるので、システムLSIの開発に係る期間をその分短縮することが可能である。

【0035】また、本発明の回路部品はハードウェア記述言語で記述されていることから、インターネットを介しての販売／流通にも適している。図9は本発明による回路部品をインターネットを介して流通させる場合の模式図である。回路部品のベンダ各社は作成した回路部品を各社のサーバに接続された記憶媒体（ハードディスク）に格納しておく。ベンダ各社が提供する回路部品を使用したい設計者は、ベンダ各社のサーバへアクセスして所定の手続の後、所望の回路部品を自分のハードディスク等へダウンロードすることにより回路部品の導入を行う。インターネットを介した流通の場合は、人手や輸送手段を介して回路部品が流通しないため、設計者は直ちに所望の回路部品を導入することができ、設計期間を従来流通にかかっていた期間分だけ短縮することができる。また、流通コストがかからないため、ベンダ側ではその分回路部品を安く提供することができ、設計者側では回路部品を安く購入することが期待できる。

【0036】尚、本発明の回路部品の販売にあたっては、回路部品本体に検証支援回路を付加して販売する他に、検証支援回路のみその回路部品本体に対応するオプション製品として回路部品本体とは別製品（別ファイル）として販売することも可能である。ただし、この場合も回路部品本体と検証支援回路との接続は回路部品の提供者によって予め接続されるよう設定されていなければならない。検証支援回路のみ別製品として販売する場合は、回路部品本体のみを購入した後に検証支援回路を実際に必要とする時点で使用者が検証支援回路を単体で購入することができる。

【0037】本発明の回路部品の販売方法によれば、回路部品の提供者によって予め回路部品本体に接続されている検証支援回路を回路部品本体に付加して、または回路部品本体に対応するオプション製品として回路部品本体とは独立に販売／流通することから、回路部品購入者における回路部品の検証を容易にし、従来に比べて回路システム開発に係る期間を短縮させることができる。

【0038】

【発明の効果】以上説明したように、本発明のシステムLSIの回路部品によれば、特定の回路機能を有する回路部品本体に対し、回路部品の提供者側によって回路部品本体と切り離し可能に接続されていると共に、回路部品本体との接続が切断された場合でも回路部品本体単体での動作に影響を与えないように回路部品本体の回路内部と接続されている検証用出力端子を設けるようにしたので、回路部品を導入した使用者側にて回路部品の動作検証が容易になると共に、論理合成する際には必要な回路部分のみ論理合成することが可能である。したがっ

て、ハードウェア記述言語レベルでの回路システムの動作検証、及び修正にかかる時間が従来よりも短縮され、システムLSIの開発、製造に係る期間をその分短縮することができるようになる。

【図面の簡単な説明】

【図1】本発明の一実施形態に係るシステムLSIの回路部品の模式図である。

【図2】本発明の一実施形態に係るシステムLSIの回路部品のハードウェア記述言語による記述の一例（1／3）である。

【図3】本発明の一実施形態に係るシステムLSIの回路部品のハードウェア記述言語による記述の一例（2／3）である。

【図4】本発明の一実施形態に係るシステムLSIの回路部品のハードウェア記述言語による記述の一例（3／3）である。

【図5】本発明の一実施形態に係るシステムLSIの回路部品を検証する際に使用される検証ツールのハードウェア記述言語による記述の一例（1／2）である。

【図6】本発明の一実施形態に係るシステムLSIの回路部品を検証する際に使用される検証ツールのハードウェア記述言語による記述の一例（2／2）である。

【図7】本発明による回路部品の導入から回路システムの回路合成までの各段階の一例を示した模式図である。

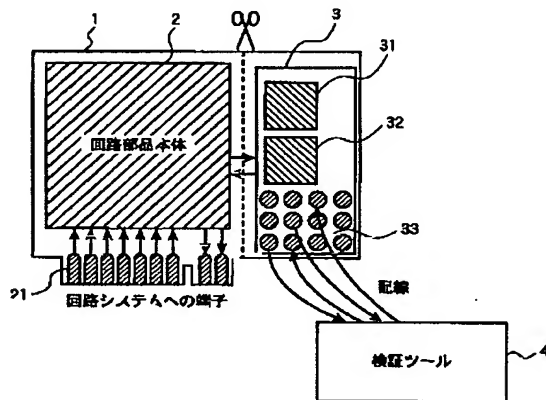
【図8】本発明による回路部品を使用したシステムLSIの製造方法のフローチャートである。

【図9】本発明による回路部品をインターネットを介して流通させる場合の模式図である。

【符号の説明】

- 1 回路部品
- 2 回路部品本体
- 21 入出力端子
- 3 検証支援回路
- 31 駆動回路
- 32 モニタ回路
- 33 検証用出力端子
- 4 検証ツール
- 5 回路システム

【図1】



【図6】

本発明の一実施形態である回路部品を検証する際に使用される検証ツール（"DebugEnv"）のハードウェア記述言語による記述例（2／2）

```

/* Status Monitor */
always @ (Compo23A, oActiveRatio < 32' h00000012) begin
    $display ("Low Active Ratio");
end

always @ (Compo23A, oConditionCode == 32' hFFFFFFF) begin
    $display ("Now Waiting");
end

```

:

【図2】

本発明の一実施形態である回路部品のハードウェア記述言語による記述例（1／3）

```

module Compo23A (sysclk, reset                                     <回路部品Compo23A>
,   in00sig, in01sig, ...
,   out00sig, out01sig, ... );
input sysclk, reset;
input in00sig, in01sig, ...;    /* Input Terminals */           ↓入出力端子の定義
output out00sig, out01sig, ...; /* Output Terminals */

wire Internal00Wire;           /* Internal Wire */              ↓内部信号線の定義
wire Internal01Wire;           /* Not described */              (詳細については非公知)
wire Internal02Wire;           /* In Specification */
wire Internal03Wire;
:
/* Circuit Discription */                                           ↓回路部品本体の記述
<省略>                                                              <省略>
:

`ifdef Debug                                                         <Debug回路ブロック>
/* Embedded Termintals Described In Specficaton */
wire [31:0] oStatusCode;      /* Output */                      ↓検証用出力端子の定義
wire [31:0] oConditionCode;
wire [31:0] oActiveRatio;
wire [31:0] oErrorCount;
wire [31:0] oWaitCount;
wire oChangeState;

reg [31:0] rPreviousData;
wire [31:0] oPreviousData;
wire oSeriousError;
wire oHald;
:
wire i00Condition;            /* Input */                        ↓入力設定項目の定義
wire i01Condition;
wire i02Condition;

```

【図 3】

本発明の一実施形態である回路部品のハードウェア記述言語による記述例 (2/3)

```

/* Connections to Circuit Component */
/* Not Described in Specification */
assign oStatusCode = Internal84Wire ;           ↓検証用出力端子と回路
assign oConditionCode = Internal107Wire ;       部品本体との接続
assign oActiveRatio = Internal84Wire / Internal12Wire * 100 ;
assign oErrorCount = Internal134Wire ;
assign oWaitCount = (Internal134Wire == 1'b0) ? Internal43Wire : Internal32Wire ;
assign oSeriousError = Internal32Wire | Internal33Wire | Internal49Wire ;
assign oChangeState = (Internal31Wire == Internal62Wire) ? 1'b0 : 1'b1 ;
always @ (oChangeState) rPreviousData : Internal22Wire ;
assign oPreviousdata = rPreviousdata ;
assign oHalt = Internal33Wire ;
:
`endif
endmodule

`ifdef StandAlone                                <StandAlone回路ブロック>

module Driver23A :                               /* Driver for Compo23A */      <駆動回路Driver23A>
reg sysclk, reset ;
reg in00sig, in01sig, . . . ;
wire out00sig, out01sig, . . . ;
/* Connection */
Compo23A Compo23A(sysclk, reset
, in00sig, in01sig, . . .
, out00sig, out01sig, . . . );
parameter CLK = 10 ;                            /* Definition of Clock Cycle */   クロック長の定義
always # (CLK/2) sysclk = ~sysclk ; /* Clock Driver */   ↓クロック生成回路の記述
Initial begin
sysclk = 1'b1 ;
end

```

【図4】

本発明の一実施形態である回路部品のハードウェア記述言語による記述例 (3/3)

```

/* Reset & Signal Generator */      ↓エミュレーション用信号
initial begin                        生成回路の記述
    reset = 1'b1;                    /* Reset */
    $readmemh (TestSignal, rSignal);
# (CLK * 10 + 2)
    reset = 1'b0;                    /* End of Reset */
    iDataNum = 0;                    /* Signal Generator */
    for (iLoopCount = 0; iLoopCount < Max_SIGNAL
        ; iLoopCount = iLoopCount + 1) begin
        rSignalTemp = rSignal[iLoopCount];
        rin00Sig = rSignalTemp[179]; /* Driving Terminals of Component */
        rin01Sig = rSignalTemp[178]; ↓入力端子へエミュレーシ
        rin02Sig = rSignalTemp[177]; ョン信号を入力する記述
        :
    end
# (CLK);
end
end
`ifdef ExternalMonitor
    Include "DebugEnv";              <ExternalMonitor回路ブロック>
    `else                             ↓外部検証ツール"DebugEnv"
                                    の読み込み
    assign Compo23A, i00Condition = 1'b0; /* Default Condition */ ↓外部検証ツール"DebugEnv"
    assign Compo23A, i01Condition = 1'b0; を読み込まない場合
    assign Compo23A, i02Condition = 1'b0;
    `endif

end module;
`endif

```

【図5】

本発明の一実施形態である回路部品を検証する際に使用される検証ツール ("DebugEnv") の
ハードウェア記述言語による記述例 (1 / 2)

```
/* Debug Environment */
```

```
/* Input Singal Data */
```

```
parameter TestSignal = "SigNo453. dat"
```

↓ Compo23Aに対して入力する付号
データ名の記述

```
/* Signal Monitor */
```

```
always @ (Compo23A, wSeriousError == 1'b1) begin
    $display ("#### Serious Error Detected!! ####");
    $finish;
end
```

↓ 回路部品本体の動作監視用モニタ
回路の記述

```
always @ (Compo23A, wHalt == 1'b1) begin
    $display ("#### Stop Because of Halt ####");
    $finish;
end
```

```
initial begin
```

```
    $monitor ("Status : %h ", Compo23A, oStatusCode
        , "Condition : %h ", Compo23A, oConditionCode
        , "Active : %h ", Compo23A, oActiveRatio
        , "Error : %h ", Compo23A, oErrorCount
        , "Wait : %h ", Compo23A, oWaitCount
```

↓ 検証用出力端子からの出力データ
を表示画面に表示させる記述

```
);
end
```

```
/* Condition Settings for Debug */
```

```
assign Compo23A, i00Condition = 1'b1;
```

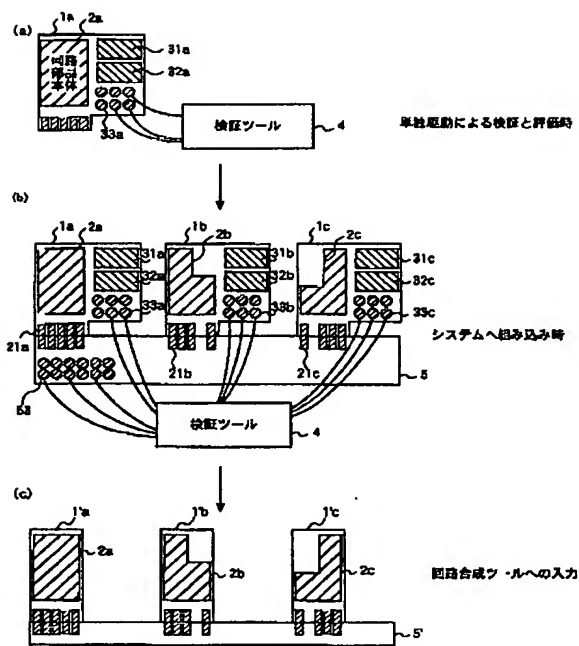
```
/* Condition */
```

```
assign Compo23A, i01Condition = 1'b0;
```

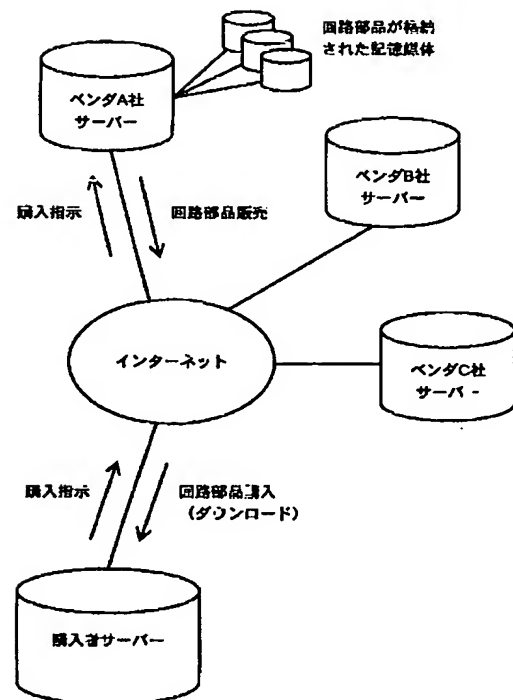
```
assign Compo23A, i02Condition = 1'b0;
```

↓ 回路部品に与える検証条件の設定

【図7】



【図9】



【図8】

